

# Statistics 341: Intro to Stat Computing and Exploratory Data Analysis with R

## Lecture 1: Introduction to R and Getting Started

Brad Mcneney

2018-01-04

Introducing R

Course objectives

Getting started

Quick-start R

# Introducing R

# What is R?

- ▶ R is an open-source environment for statistical computing and graphics.
  - ▶ An implementation of the S language  
([https://en.wikipedia.org/wiki/S\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/S_(programming_language)))
- ▶ Started in the mid-1990's by Ross Ihaka and Robert Gentleman at Auckland University
- ▶ Now maintained by a team of experts called the R Development Core Team
- ▶ A “packages” system allows any user to bundle R code, data and examples together.
- ▶ R and R packages are distributed through the Comprehensive R Archive Network (CRAN).
- ▶ SFU has a CRAN mirror at <http://cran.stat.sfu.ca>

# What does “environment” mean?

- ▶ R is a fully-functioning programming environment with all the usual constructs, such as
  - ▶ conditionals (if-then-else),
  - ▶ loops
  - ▶ user-defined functions.
- ▶ In addition there are built-in facilities for
  - ▶ data input, storage, manipulation, and output
  - ▶ optimization, matrix computation, etc.,
  - ▶ random number generation,
  - ▶ data analysis and graphics.
- ▶ “Base” R is good, but it is the package system that makes R great.

# R packages

- ▶ The R package system is the key to R's success
- ▶ It has allowed statisticians and other data scientists to implement and distribute their work to be used by others.
- ▶ The R package system enforces some rules about how packages are structured, but differences in programming styles of package authors mean different interfaces
  - ▶ Users need to be aware of different data structures for input, output and of different styles of graphics

## Course objectives

# Course objectives

- ▶ Understand basic R data structures and programming
- ▶ Learn how to use base R and R package functions for data management, exploration, presentation and analysis
- ▶ Learn how to use packages from the “tidyverse”, a collection of modern tools for data science.
  - ▶ <https://www.tidyverse.org/>



## Getting started

## Getting started with R, RStudio and git

- ▶ A brief set of instructions are on the canvas page <https://canvas.sfu.ca/courses/37593/pages/getting-started-with-r-rstudio-and-git>
- ▶ Please try to get R and RStudio installed and create an RStudio project linked to the class GitHub repository as soon as possible.
- ▶ Those still having trouble after the weekend should attend one of the Wednesday tutorials (12:30-2:20) for help, even if these are not your tutorial times.

## R reference cards

- ▶ RStudio has created several useful reference cards, called cheat sheets, and have collected cheat sheets from other R users.
- ▶ See <https://www.rstudio.com/resources/cheatsheets/>
- ▶ For getting started with RStudio you might find the following helpful: <https://github.com/rstudio/cheatsheets/raw/master/rstudio-ide.pdf>

# Starting R

- ▶ Start R by starting RStudio.
- ▶ The “Console” window is where you can type your commands.
- ▶ However, it is good practice to open an R script, type your commands in the script, and then submit the commands to the R console.
  - ▶ Session -> Set Working Directory to set the working directory
  - ▶ File -> New File -> R Script to open a new R script
  - ▶ type your commands into the script
  - ▶ put your cursor on the line you want to submit and hit Ctrl-enter
- ▶ Save your script for later use.
- ▶ More on the RStudio interface at <https://support.rstudio.com/hc/en-us/sections/200107586-Using-RStudio>

## Quick-start R

# R as a calculator

```
2+4
```

```
## [1] 6
```

```
5*6
```

```
## [1] 30
```

```
x<-6
```

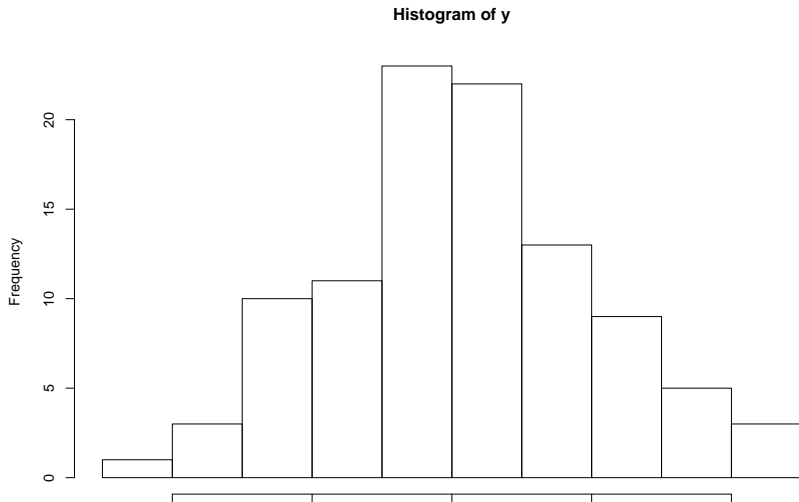
```
x*5
```

```
## [1] 30
```

- ▶ The assignment operator `<-` assigns the value 6 to the variable `x`

# Random number generation

```
set.seed(123) # optional, but makes computation reproducible  
n<-100  
y<-rnorm(n,sd=1)  
hist(y)
```



# R data

```
data(cars) # load a built-in example dataset, help(cars) for info
head(cars) # look at the first few rows
```

```
##      speed dist
## 1         4     2
## 2         4    10
## 3         7     4
## 4         7    22
## 5         8    16
## 6         9    10
```

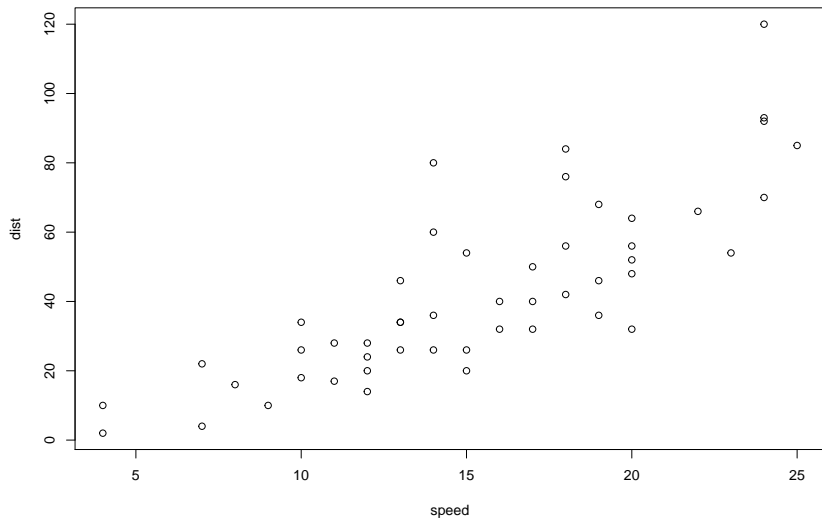
```
summary(cars) # summarize the variables
```

```
##           speed           dist
##  Min.      : 4.0    Min.      : 2.00
##  1st Qu.:12.0    1st Qu.: 26.00
##  Median :15.0    Median   : 36.00
##  Mean   :15.4    Mean     : 42.98
##  3rd Qu.:19.0    3rd Qu.: 56.00
##  Max.    :25.0    Max.     :120.00
```



# Plotting data

```
plot(cars)
```



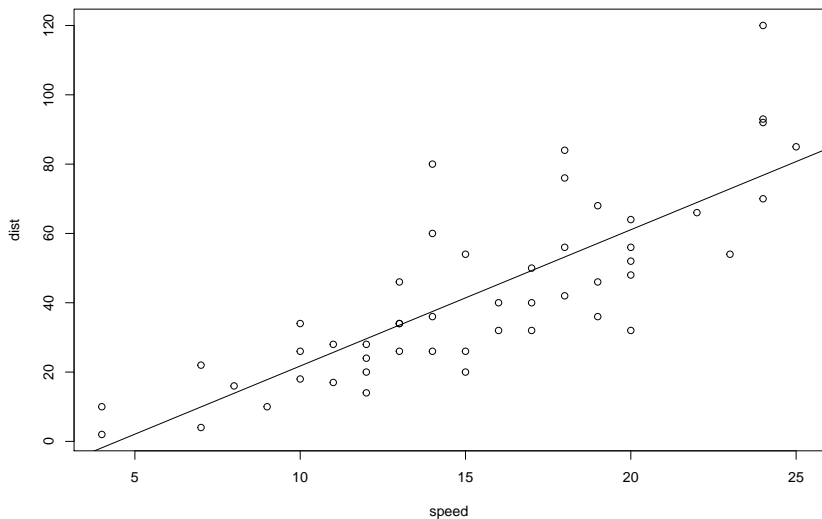
# Modelling

```
fit <- lm(dist~speed,data=cars) # fit a linear model, help(lm)
summary(fit)
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## speed        3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

## Modelling, continued

```
plot(cars)  
abline(fit)
```



# Extracting model components

```
head(residuals(fit))
```

```
##           1           2           3           4           5           6
## 3.849460 11.849460 -5.947766 12.052234  2.119825 -7.812584
```

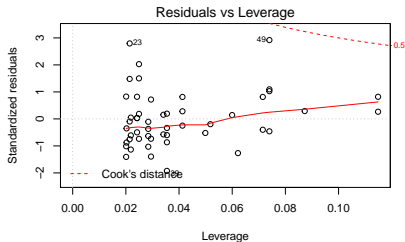
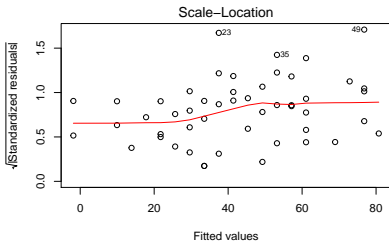
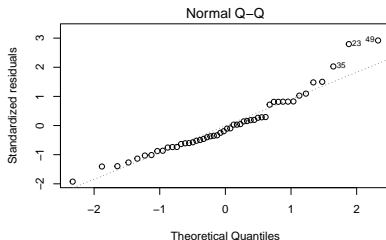
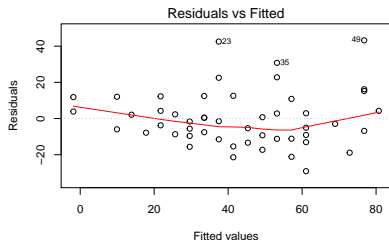
```
vcov(fit)
```

```
##           (Intercept)      speed
## (Intercept)  45.676514 -2.6588234
## speed       -2.658823  0.1726509
```

- ▶ See the help page for more components

# Model diagnostics

```
par(mfrow=c(2,2))  
plot(fit)
```



# Reading

- ▶ R and RStudio help pages/reference cards mentioned in this lecture